

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-068862

(43)Date of publication of application : 03. 03. 2000

(51)Int. Cl. H03M 13/23  
G06F 11/10

(21)Application number : 10-232580 (71)Applicant : FUJITSU LTD

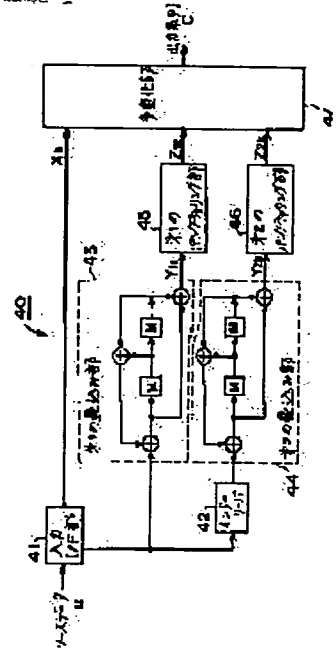
(22)Date of filing : 19. 08. 1998 (72)Inventor : RI KEIHO

## (54) ERROR CORRECTION CODER

## (57)Abstract:

PROBLEM TO BE SOLVED: To obtain a desired coding rate in the error correction coder that is provided with a plurality of convolution circuits connected in parallel with each other.

SOLUTION: Convolution sections 43, 44 placed in parallel respectively execute convolution processing for source data (u). An interleaver 42 randomizes the source data (u) given to the convolution section 44. Then puncturing sections 45, 46 select a data element from parity data series Y1K, Y2K according to a predetermined puncturing pattern respectively. A multiplexer section 47 imparts data elements selected by the puncturing sections 45, 46 to the data series XK.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's  
decision of rejection][Kind of final disposal of application  
other than the examiner's decision of  
rejection or application converted  
registration][Date of final disposal for  
application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's  
decision of rejection][Date of requesting appeal against  
examiner's decision of rejection]

[Date of extinction of right]

Copyright (C) ; 1998, 2000 Japan Patent Office



(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-68862

(P2000-68862A)

(43) 公開日 平成12年3月3日 (2000.3.3)

(51) Int.Cl.<sup>7</sup>

識別記号

F I

テラート\* (参考)

H 0 3 M 13/23

H 0 3 M 13/12

G 0 6 F 11/10

3 3 0

G 0 6 F 11/10

3 3 0 N

審査請求 未請求 請求項の数18 O L (全 23 頁)

(21) 出願番号 特願平10-232580

(22) 出願日 平成10年8月19日 (1998.8.19)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 発明者 李 維峰

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74) 代理人 100074099

弁理士 大菅 義之 (外1名)

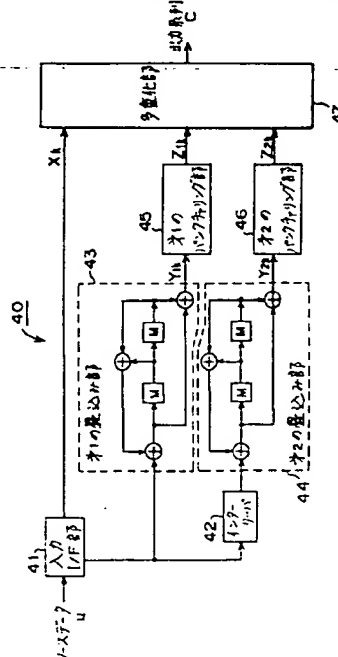
(54) 【発明の名称】 誤り訂正符号化装置

(57) 【要約】

【課題】 互いに並列に設けられた複数の畳込み回路を備える誤り訂正符号化装置において所望の符号化率が得られるようにする。

【解決手段】 互いに並列に設けられた畳込み部43および44は、それぞれソースデータuに対して畳込み処理を実行する。インターリーバ42は、畳込み部44に対して与えられるソースデータuをランダム化する。バンクチャリング部45および46は、それぞれ予め決められたバンクチャリングパターンに従ってパリティデータ系列Y1KおよびY2Kからデータエレメントを選択する。多重化部47は、データ系列XKに対してバンクチャリング部45および46により選択されたデータエレメントを付与する。

本発明の一実施形態の誤り訂正符号化装置の構成図



## 【特許請求の範囲】

【請求項1】 ソースデータを符号化する誤り訂正符号化装置であって、

互いに並列に設けられた複数の畳込み手段と、

上記複数の畳込み手段に対して互いに異なるデータ系列が与えられるように上記ソースデータをランダム化するランダム化手段と、

上記複数の畳込み手段の各出力の各データエレメントを選択するか否かを表し且つそのデータ長が上記複数の畳込み手段の各出力のデータ長と同じである選択情報に従って、対応する畳込み手段の出力のデータエレメントを選択する選択手段と、

上記ソースデータおよび上記選択手段により選択されたデータエレメントを出力する出力手段と、

を有する誤り訂正符号化装置。

【請求項2】 上記選択情報が上記複数の畳込み手段に対してそれぞれ設定される請求項1に記載の誤り訂正符号化装置。

【請求項3】 上記複数の畳込み手段に対してそれぞれ設定される選択情報が互いに同じである請求項1に記載の誤り訂正符号化装置。

【請求項4】 上記複数の畳込み手段の各出力からそれぞれ選択するデータエレメントの数が互いに同じである請求項1に記載の誤り訂正符号化装置。

【請求項5】 上記選択手段は、上記複数の畳込み手段の各出力の各先頭データエレメントを選択しない請求項1に記載の誤り訂正符号化装置。

【請求項6】 上記選択手段は、上記複数の畳込み手段の各出力からそれぞれデータエレメントを均等に選択する請求項1に記載の誤り訂正符号化装置。

【請求項7】 上記複数の選択情報が書換え可能な記憶装置に格納されている請求項1に記載の誤り訂正符号化装置。

【請求項8】 Nビットのソースデータを符号化する誤り訂正符号化装置であって、

互いに並列に設けられ、それぞれNビットの入力系列に対してNビットの出力系列を生成する複数の畳込み手段と、

上記複数の畳込み手段に対して互いに異なる系列が与えられるように上記ソースデータをランダム化するランダム化手段と、

上記複数の畳込み手段により生成される各出力系列の各データエレメントを選択するか否かを表すNビットの選択情報を格納する格納手段と、

上記選択情報に従って上記複数の畳込み手段により生成された出力系列のデータエレメントを選択する選択手段と、

上記ソースデータおよび上記選択手段により選択されたデータエレメントを出力する出力手段と、

を有する誤り訂正符号化装置。

【請求項9】 Nビットのソースデータを符号化する誤り訂正符号化装置であって、

上記ソースデータに対してNビットのデータ系列を生成する第1の畳込み手段と、

上記ソースデータをランダム化するランダム化手段と、上記第1の畳込み手段に対して並列に設けられ、上記ランダム化手段の出力に対してNビットのデータ系列を生成する第2の畳込み手段と、

上記第1の畳込み手段により生成されるデータ系列の各データエレメントを選択するか否かを表すNビットの選択情報に従って上記第1の畳込み手段により生成されるデータ系列のデータエレメントを選択する第1の選択手段と、

上記第2の畳込み手段により生成されるデータ系列の各データエレメントを選択するか否かを表すNビットの選択情報に従って上記第2の畳込み手段により生成されるデータ系列のデータエレメントを選択する第2の選択手段と、

上記ソースデータおよび上記第1および第2の選択手段により選択されたデータエレメントを出力する出力手段と、

を有する誤り訂正符号化装置。

【請求項10】 ソースデータを符号化するための誤り訂正符号化装置であって、

互いに並列に設けられた複数のエンコーダと、

上記複数のエンコーダに対して互いに異なるデータ系列が与えられるように上記ソースデータをランダム化するインターリーバと、

上記複数のエンコーダの各出力の各データエレメントを選択するか否かを表す選択情報に従って上記複数のエンコーダの出力のデータエレメントを選択する選択手段と、

上記ソースデータおよび上記選択手段により選択されたデータエレメントを出力する出力手段と、

を有し、上記選択情報のデータ長が上記ソースデータを上記複数のエンコーダの中の1つに入力したときに得られる出力データ系列のデータ長と同じであることを特徴とする誤り訂正符号化装置。

【請求項11】 ソースデータを符号化するための誤り訂正符号化装置であって、

要求される符号化率に応じて上記ソースデータの中の所定数のデータエレメントを重複させる重複手段と、

互いに並列に設けられた複数の畳込み回路を備え、上記重複手段により所定数のデータエレメントが重複させられたソースデータに対して符号化処理を行う符号化回路と、

を有する誤り訂正符号化装置。

【請求項12】 上記重複手段は、上記畳込み回路のメモリに格納されるビット数に1を加えたデータ長ごとにデータエレメントを重複させる請求項11に記載の誤り

訂正符号化装置。

【請求項 13】 ソースデータを符号化するための誤り訂正符号化装置であって、

要求される符号化率に応じて上記ソースデータに対して所定数のダミービットを挿入する挿入手段と、

互いに並列に設けられた複数の畳込み回路を備え、上記挿入手段によりダミービットが挿入されたソースデータに対して符号化処理を行う符号化回路と、

を有する誤り訂正符号化装置。

【請求項 14】 上記ダミービットの値が 1 または 0 である請求項 13 に記載の誤り訂正符号化装置。

【請求項 15】 上記挿入手段は、上記畳込み回路のメモリに格納されるビット数と同じビット数を 1 単位として上記ソースデータにダミービットを挿入する請求項 13 に記載の誤り訂正符号化装置。

【請求項 16】 符号化データを含む無線信号が伝送される無線通信システムにおいて移動機との間で上記無線信号を送受信する基地局装置であって、

互いに並列に設けられ、移動機へ送信すべきソースデータに対してそれぞれ畳込み処理を行う複数の畳込み回路と、

上記複数の畳込み回路に対して互いに異なるデータ系列が与えられるように上記ソースデータをランダム化するインターリーブと、

上記複数の畳込み回路の各出力の各データエレメントを選択するか否かを表し且つそのデータ長が上記複数の畳込み回路の各出力のデータ長と同じである選択情報に従って、対応する畳込み回路の出力のデータエレメントを選択する選択手段と、

上記ソースデータおよび上記選択手段により選択されたデータエレメントを出力する出力手段と、を含む誤り訂正符号化装置を備えた基地局装置。

【請求項 17】 符号化データを含む無線信号が伝送される無線通信システムにおいて移動機との間で上記無線信号を送受信する基地局装置であって、

要求される符号化率に応じて上記移動機へ送信すべきソースデータの中の所定数のデータエレメントを重複させる重複手段と、

互いに並列に設けられた複数の畳込み回路を備え、上記重複手段により所定数のデータエレメントが重複させられたソースデータに対して符号化処理を行う符号化回路と、を有する誤り訂正符号化装置を備えた基地局装置。

【請求項 18】 符号化データを含む無線信号が伝送される無線通信システムにおいて移動機との間で上記無線信号を送受信する基地局装置であって、

要求される符号化率に応じて上記移動機に送信すべきソースデータに対して所定数のダミービットを挿入する挿入手段と、

互いに並列に設けられた複数の畳込み回路を備え、上記挿入手段によりダミービットが挿入されたソースデータ

に対して符号化処理を行う符号化回路と、を有する誤り訂正符号化装置を備えた基地局装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、符号化装置に係わり、特に、誤り訂正符号に係わる。

【0002】

【従来の技術】符号化技術は、様々な分野において広く利用されている。たとえば、データ伝送においては、送信元装置が伝送すべきデータを符号化して通信路に送出し、受信装置がその符号化されているデータを受信して復号する。また、データを記憶装置に保存する際には、データは、符号化されてディスクなどに書き込まれる。そして、符号化されているデータは、そのディスクから読み出される際に復号される。なお、符号化とは、一般には、情報源データ系列を異なるデータ系列に変換することをいい、また、その変換によって得られる新しいデータ系列を符号という。

【0003】ところで、符号化されたデータを伝送する際、その通信路においてエラーが発生することがある。また、符号化されたデータを格納する記憶装置からそのデータを読み出して再生する際にも、エラーが発生し得る。このようなエラーの発生を検出するため、或いはそのエラーを訂正するために、誤り訂正符号がしばしば使用されている。

【0004】誤り訂正符号の 1 つとして、畳込み符号が知られている。畳込み符号では、 $n$  ビットのデータが入力される毎に、その  $n$  ビットのデータ及びその  $n$  ビットのデータの直前に入力された  $s$  ビットのデータに従って決まる  $m$  ( $m > n$ ) ビットのデータが出力される。すなわち、畳込み符号では、伝送すべきデータに対して、「 $m - n$ 」ビットのデータが誤り訂正のために付与される。このことにより、データの冗長性が増加するので、復号時の復号誤り率を低くすることができる。

【0005】なお、伝送すべきデータの量（ソースデータのビット数）と、符号化によって得られるデータの量（出力データのビット数）との比率は、一般に符号化率（または、情報率） $R$  と呼ばれ、下式により与えられる。

【0006】 $R = n/m$

この符号化率  $R$  は、誤り訂正符号においては、常に 1 よりも小さい。また、一般に、符号化率  $R$  は、復号装置における誤り訂正能力を決めるパラメータの 1 つである。たとえば、符号化率  $R$  が小さいと、誤り訂正能力は高くなる。

【0007】図 20 は、畳込み符号を用いた既存の誤り訂正符号化装置の一例のブロック図である。この誤り訂正符号化装置 500 は、互いに並列に設けられた 2 つの畳込み部 501 および 502 を備える。このように、互いに並列に設けられた複数の畳込み部を備える符号化装

置は、しばしば、「ターボ符号化装置」と呼ばれている。

【0008】誤り訂正符号化装置500は、ソースデータdに対して、データ系列x、及びそのデータ系列xを訂正するためのパリティデータ系列y1、y2を生成する。データ系列x、パリティデータ系列y1、y2は、多重化されて出力される。そして、この出力がソースデータdの符号化データである。以下では、Nビットのソースデータdを符号化する際の動作を説明する。

【0009】ソースデータdは、そのままデータ系列xとして出力されるとともに、畳込み部501およびインターリーバ503に与えられる。畳込み部501は、ソースデータdに対して畳込み符号化を実行してパリティデータ系列y1を出力する。インターリーバ503は、入力されたNビットのソースデータdをいったん保持し、その後に入力順序と異なる順序でその保持されているソースデータdを読み出して出力する。このことにより、ソースデータdはランダム化される。インターリーバ503の出力は、畳込み部502に対して与えられる。そして、畳込み部502は、そのインターリーバ503の出力に対して畳込み符号化を実行してパリティデータ系列y2を出力する。

【0010】上記動作により、誤り訂正符号化装置500は、Nビットのソースデータdに対して、Nビットのデータ系列x、Nビットのパリティデータ系列y1、およびNビットのパリティデータ系列y2を生成する。これらのデータ系列x、パリティデータ系列y1、y2

は、例えば、ビット毎に多重化されて符号化データとして出力される。従って、この場合、誤り訂正符号化装置500は、Nビットの入力の対して3×Nビットのデータを出力するので、その符号化率Rは、1/3となる。

【0011】図21は、図20に示した誤り訂正符号化装置の変形例のブロック図である。誤り訂正符号化装置510は、図20に示した誤り訂正符号化装置500に対して選択部511を設けることにより実現される。選択部511は、予め決められている選択パターンに従って、畳込み部501および502によりそれぞれ生成されるパリティデータ系列y1およびy2を選択してパリティデータ系列Zとして出力する。尚、選択部511の動作は、「パンクチャリング(Puncturing)」と呼ばれることがある。

【0012】選択部511は、畳込み部501および502の出力を交互に1ビットずつ選択する。選択部511の出力系列Zを表1に示す。ここで、「y1(i)」は、ソースデータdの第i番目のデータエレメントに対応する畳込み部501の出力であり、「y2(i)」は、ソースデータdの第i番目のデータエレメントに対応する畳込み部502の出力である。このように、選択部511は、誤り訂正符号化装置510にNビットのソースデータdが入力されると、Nビットの出力系列Z(y1(1), y2(2), y1(3), y2(4), ..., y1(N-1), y2(N))を出力する。

【0013】

【表1】

y <sub>1</sub> (1)		y <sub>1</sub> (3)		.....	y <sub>1</sub> (N-1)	
	y <sub>2</sub> (2)		y <sub>2</sub> (4)	.....		y <sub>2</sub> (N)

【0014】選択部511によるパンクチャリング動作は、下記に示す式で表すことができる。

【0015】

【数1】

$$Z = D \cdot P$$

$$= \begin{bmatrix} y_1(1) & y_2(1) \\ y_1(i+1) & y_2(i+1) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(i = 1, 3, 5, ..., N-1)

【0016】出力系列Zは、データ行列Dとパンクチャリング行列Pとの掛け算により得られる。例えば、ソースデータdの第i番目のデータエレメントに対しては、データ行列Dの第1行とパンクチャリング行列Pの第1列との積により「y1(i)」が得られる。また、ソースデータdの第i+1番目のデータエレメントに対しては、データ行列Dの第2行とパンクチャリング行列Pの第2列との積により「y2(i+1)」が得られる。したが

って、選択部511が畳込み部501および502の出力を交互に1ビットずつ選択する動作は、上記演算を繰り返し実行する動作として表される。

【0017】上記構成により、誤り訂正符号化装置510は、Nビットのソースデータdに対して、Nビットのデータ系列x、およびNビットのパリティデータ系列Zを生成する。これらのデータ系列xおよびパリティデータ系列Zは、ビット毎に多重化されて符号化データとして出力される。従って、この場合、誤り訂正符号化装置510は、Nビットの入力の対して2×Nビットのデータを出力するので、その符号化率Rは、1/2となる。

【0018】なお、図20および図21に示した誤り訂正符号化装置については、米国特許5,446,747に詳しく開示されている。

【0019】

【発明が解決しようとする課題】ところで、ソースデータdのデータ長N(ビット数)に対する符号化装置の出力系列のデータ長Mを任意の長さにした、という要求

がある。たとえば、移動体通信システムにおいては、一般に、所定のデータ長ごとに分割された音声データ等が予め決められたデータ長のフレームの中に格納されて伝送される。すなわち、移動体通信システムにおいて符号を使用する場合には、所定のデータ長ごとに分割された音声データ等は、符号化された後にフレームの中に格納される。

【0020】ところが、図20または図21に示した従来の誤り訂正符号化装置の符号化率Rはそれぞれ固定されていた。したがって、予め決められた固定長のデータ（上述の例では、フレーム）を作成する場合、そのフレームのデータ格納領域を満たすためには、そこに無意味な情報を格納しなければならなかった。

【0021】図22(a)は、図20に示す誤り訂正符号化装置500を使用してソースデータを符号化して固定長のフレームに格納する場合の処理を説明する図である。ここでは、ソースデータdが333ビット、また、フレームのデータ格納領域が1500ビットである。この場合、誤り訂正符号化装置500は、333ビットのデータ系列x、333ビットのパリティデータ系列y<sub>1</sub>、および333ビットのパリティデータ系列y<sub>2</sub>を生成するので、フレームのデータ格納領域を満たすためには、図22(b)に示すように、501ビットのダミーデータをそのフレームに格納しなければならない。従って、網を介してそのフレームを伝送すると、無駄なデータが伝送されることになり、網資源が浪費されてしまう。

【0022】図23(a)は、図21に示す誤り訂正符号化装置510を使用してソースデータを符号化して固定長のフレームに格納する場合の処理を説明する図である。ここでは、ソースデータdが666ビット、フレームのデータ格納領域が1500ビットである。この場合、選択部511は、バンクチャリング処理により、パリティデータ系列y<sub>1</sub>およびパリティデータ系列y<sub>2</sub>からパリティデータ系列Zを生成する。したがって、誤り訂正符号化装置510は、666ビットのデータ系列x、および666ビットのパリティデータ系列Zを生成するので、フレームのデータ格納領域を満たすためには、図23(b)に示すように、168ビットのダミーデータをそのフレームに格納しなければならない。したがって、図22に示した例と同様に、無駄なデータが伝送されることになる。

【0023】このように、互いに並列に設けられた複数の畳込み部を備える既存の誤り訂正符号化装置は、その符号化率を所望の値に設定できなかったため、ソースデータを符号化して所定のフレームに格納する際、その効率が悪かった。

【0024】本発明の課題は、互いに並列に設けられた複数の畳込み部を備える誤り訂正符号化装置において所望の符号化率が得られるようにすることである。

【0025】

【課題を解決するための手段】この発明の誤り訂正符号化装置は、互いに並列に設けられた複数の畳込み手段と、上記複数の畳込み手段に対して互いに異なるデータ系列が与えられるように上記ソースデータをランダム化するランダム化手段と、上記複数の畳込み手段の各出力の各データエレメントを選択するかどうかを表し且つそのデータ長が上記複数の畳込み手段の各出力のデータ長と同じである選択情報に従って対応する畳込み手段の出力のデータエレメントを選択する選択手段と、上記ソースデータおよび上記選択手段により選択されたデータエレメントを出力する出力手段と、を有する。

【0026】上記構成において、各畳込み手段は、ソースデータを訂正するためのデータエレメントを生成する。選択手段は、上記複数の畳込み手段により生成されるデータエレメントの中から選択情報により指示されるデータエレメントを出力する。したがって、出力手段により出力される符号化データのビット数は、上記選択情報に基づいて決まる。すなわち、上記選択情報に従って所望の符号化率を得ることができる。

【0027】本発明の他の形態の誤り訂正符号化装置は、要求される符号化率に応じてソースデータの中の所定数のデータエレメントを重複させる重複手段と、互いに並列に設けられた複数の畳込み回路を備え、上記重複手段により所定数のデータエレメントが重複させられたソースデータに対して符号化処理を行う符号化回路と、を有する。

【0028】上記構成において、データエレメントを重複させる数を変えることにより、ソースデータのデータ長と上記符号化回路の出力データのデータ長との比率が変わる。即ち、符号化率が変わる。また、データエレメントが重複されていると、復号特性が向上する。

【0029】本発明のさらに他の誤り訂正符号化装置は、要求される符号化率に応じてソースデータに対して所定数のダミービットを挿入する挿入手段と、互いに並列に設けられた複数の畳込み回路を備え、上記挿入手段によりダミービットが挿入されたソースデータに対して符号化処理を行う符号化回路と、を有する。

【0030】上記構成において、挿入すべきダミービット数を変えることにより、ソースデータのデータ長と上記符号化回路の出力データのデータ長との比率が変わる。即ち、符号化率が変わる。また、予め決められているダミービット（例えば、1）が挿入されていると、復号特性が向上する。

【0031】

【発明の実施の形態】本発明の誤り訂正符号化装置は、様々な分野に適用可能であり、たとえば、通信システムや、データ記憶装置において利用され得る。

【0032】図1は、本実施形態の誤り訂正符号化装置が適用される移動体通信システムの構成図である。無線

方式は、例えば、CDMAである。基地局10は、移動機20へ送出すべきデータ（データA）を符号化するエンコーダ11、符号化されたデータを変調する変調器12、および変調されたデータを送信する送信機13を備える。基地局10から送出された無線信号は、移動機20の受信機21により受信され、復調器22により復調され、そしてデコーダ23により復号される。また、基地局10は、移動機20から送られてくる信号を受信する受信機14、受信信号を復調する復調器15、および復調されたデータを復号するデコーダ16を備える。なお、移動機20は、基地局10へ送出すべきデータ（データB）をエンコーダ24を用いて符号化し、変調器25を用いて変調し、そして送信機26を用いて送信する。

【0033】上記通信システムにおいて、本実施形態の誤り訂正符号化装置は、基地局10に設けられるエンコーダ11または移動機20に設けられるエンコーダ24に相当する。

【0034】図2は、本実施形態の誤り訂正符号化装置が適用される記憶装置の構成図である。この記憶装置30は、データ格納部33に書き込むべきデータを符号化するエンコーダ31、符号化されたデータをデータ格納部33に書き込む書き込み制御部32を備える。データ格納部33は、たとえば、光ディスク、磁気ディスク、半導体メモリなどの記憶媒体を含む。また、記憶装置30は、データ格納部33からデータを読み出す読出し制御部34、および読み出されたデータを復号するデコーダ35を備える。

【0035】上記記憶装置において、本実施形態の誤り訂正符号化装置は、エンコーダ31に相当する。図3は、本発明の一実施形態の誤り訂正符号化装置のブロック図である。この誤り訂正符号化装置の基本的な構成は、図2.1に示した従来の誤り訂正符号化装置と同じである。本実施形態の誤り訂正符号化装置のバンクチャリング部は、図2.1に示した従来の誤り訂正符号化装置の選択部511に対応する。但し、このバンクチャリング部は、選択部511とは異なる機能を備える。本実施形態の誤り訂正符号化装置は、バンクチャリング処理により所望の符号化率を実現する。以下、本実施形態の誤り訂正符号化装置の構成および動作について説明する。

【0036】誤り訂正符号化装置40は、組織符号を用いてソースデータuを符号化する。組織符号では、伝送すべきデータと、そのデータを伝送する際に発生するエラーを訂正するためのデータ（以下では、「パリティデータ」と呼ぶ）とが互いに分離されている。すなわち、誤り訂正符号化装置40は、ソースデータuを受け取ると、そのソースデータuにパリティデータzを付加して出力する。なお、誤り訂正符号化装置40は、Nビットのソースデータuを符号化する。また、誤り訂正符号化装置40は、ソースデータuを「データ系列Xk」とし

て出力する。パリティデータを「パリティデータ系列Zk」として出力する。

【0037】入力I/F部41は、受信したソースデータuを多重化部47、第1の畳込み部43、およびインターリーバ42に与える。なお、入力I/F部41から多重化部47に与えられるソースデータuのことを「データ系列Xk」と呼ぶ。

【0038】インターリーバ42は、入力されたソースデータuをランダム化する。インターリーバ42は、Nビットのソースデータuを一時的に保持するためのメモリを備える。入力されたNビットのソースデータuは、1ビットずつそのメモリに書き込まれる。そして、そのメモリに書き込まれたデータは、そのメモリへの書き込み順序とは異なる順序で1ビットずつ読み出される。このことにより、ソースデータuはランダム化される。

【0039】なお、インターリーバ42を設けた目的は、畳込み部43および44に対して互いに異なる独立したデータ系列を与えることである。したがって、図3においては、第2の畳込み部44の前段にのみインターリーバを設けているが、第1の畳込み部43および第2の畳込み部44に対してそれぞれインターリーバを設けてもよい。ただし、その場合には、それら2つのインターリーバにより実行されるランダム化処理が互いに異なっている必要がある。

【0040】第1の畳込み部43は、入力されたソースデータuに対して畳込み処理を実行する。一方、第2の畳込み部44は、インターリーバ42によってランダム化されたソースデータuに対して畳込み処理を実行する。第1の畳込み部43および第2の畳込み部44は、互いに同じ構成であってもよいし、互いに異なる構成であってもよい。以下では、これら2つの畳込み部43、44の構成が互いに同じであるものとする。

【0041】第1の畳込み部43は、互いに直列に接続された複数のメモリM、および1つ以上の加算機を備える。各メモリMは、たとえば、フリップフロップであり、それぞれ1ビットのデータを保持する。直列に接続されたメモリMは、シフトレジスタを構成する。加算機は、例えば、排他論理和演算器、又はmod 2加算機である。図3に示す構成では、第1の畳込み部43は、2つのメモリM、および3つの加算機を備える。この場合、メモリMにより保持されるデータ量が2ビットなので、拘束長=2である。なお、この明細書では、畳込み回路の拘束長=畳込み回路のメモリに格納されるデータのビット数とする。

【0042】第1の畳込み部43は、ソースデータuのデータエレメントを受け取る毎に、その受け取ったデータエレメントに対応するパリティデータ系列Yikのデータエレメントを出力する。パリティデータYikのデータエレメントは、第1の畳込み部43へ新たに入力されたデータエレメントと、そのデータエレメントが入力され



たタイミングにおいて上記メモリMに保持されているデータエレメントとの和によって得られる。すなわち、この畳込み処理では、新たに入力されたデータエレメントに対応するデータエレメントは、先に入力されている1以上のデータエレメントおよびその新たに入力されたデータエレメントに基づいて生成されて出力される。

【0043】第1の畳込み部43の各メモリMには、それぞれ初期値として「0」が設定される。そして、第1の畳込み部43は、Nビットのデータ系列が入力されると、Nビットのパリティデータ系列を出力し、さらにそのパリティデータ系列に続いて終結ビット (tail bit) を出力する。終結ビットのデータ長は、例えば、メモリMの数と同じであり、「2」である。

【0044】第2の畳込み部44の構成および動作は、基本的に上述した第1の畳込み部43の構成および動作と同じである。但し、第2の畳込み部44は、インターリーバ42によってランダム化されたソースデータuに対して畳込み処理を実行し、パリティデータ系列Y2kを生成する。

【0045】なお、畳込み処理は、既知の技術であり、当業者にはよく知られているので、ここではその詳細な説明は省略する。第1のバンクチャリング部45は、第1の畳込み部43により生成されるパリティデータ系列Y1kの各データエレメントを予め決められたパターンに従って選択し、パリティデータ系列Z1kとして出力する。同様に、第2のバンクチャリング部46は、第2の畳込み部44により生成されるパリティデータ系列Y2kの各データエレメントを予め決められたパターンに従って選択し、パリティデータ系列Z2kとして出力する。図3に示す畳込み符号化装置40の特徴は、これらのバンクチャリング部によるデータエレメントの選択方法である。データエレメントの選択方法については、後述詳しく説明する。

【0046】多重化部47は、入力I/F部41から与えられるデータ系列Xk、第1のバンクチャリング部45から与えられるパリティデータ系列Z1k、および第2のバンクチャリング部46から与えられるパリティデータ系列Z2kを多重化して出力する。この多重化部47からの出力系列Cは、ソースデータuに対する符号化データである。なお、多重化部47は、入力される3つのデータ系列のタイミングを調整する機能を備えている。これにより、ソースデータu (データ系列Xk) の各データエレメントが出力される際には、そのソースデータuのデータエレメントに対応するパリティデータ系列Z1kおよびZ2kの各データエレメントがそのソースデータのデータエレメントに関連づけられて出力される。

【0047】このように、誤り訂正符号化装置40は、ソースデータuが入力されると、そのソースデータuのデータ系列と同じデータ系列であるデータ系列Xkに、誤り訂正のためのパリティデータ系列Z1kおよびZ2kを

付与して出力する。

【0048】次に、第1のバンクチャリング部45および第2のバンクチャリング部46の動作を構成および説明する。ここでは、ソースデータuのデータ長がNビットであり、出力系列Cのデータ長がMビットであることが予め決められているものとする。この場合、誤り訂正符号化装置40は、符号化率 $=N/M$ が要求される。ソースデータuおよび出力系列Cのデータ長は、たとえば、この誤り訂正符号化装置が通信システムにおいて利用される場合には、その通信システムの仕様により決定される。特に、出力系列Cのデータ長は、その通信システムで伝送されるフレームのフォーマットにより決められる。

【0049】図4は、第1のバンクチャリング部45のブロック図である。なお、第2のバンクチャリング部46も基本的に同じ構成である。ラッチ回路51は、第1の畳込み部43から出力されるパリティデータ系列Y1kを1ビットずつ保持する。すなわち、ラッチ回路51は、第1の畳込み部43からパリティデータ系列Y1kのデータエレメントが出力される毎に更新される。CPU52は、メモリ53に格納されているプログラムを実行することにより、ラッチ回路51に保持されているデータエレメントからパリティデータZ1kのデータエレメントを生成する。パリティデータZ1kデータエレメントは、出力ポート54を介して多重化部47へ送出される。

【0050】メモリ53は、CPU52により実行されるプログラム、およびそのプログラムにより利用されるバンクチャリングテーブルを格納する。このプログラムについては、後述説明する。

【0051】図5は、バンクチャリングテーブルの一例を示す図である。バンクチャリングテーブルは、パリティデータ系列Y1kの各データエレメントを選択するか否かを表す選択情報 (バンクチャリングパターン情報) を格納する。したがって、この選択情報のデータ長は、第1の畳込み部43からの出力データ系列のデータ長と同じになる。第1の畳込み部43は、ソースデータuのデータ長がNビットである場合には、Nビットのパリティデータ系列Z1kを出力する。したがって、この場合、選択情報のデータ長も、Nビットになる。

【0052】なお、第1の畳込み部43は、ソースデータuを受信すると、パリティデータ系列Y1kを出力し、その後終結ビットを出力する。ただし、この終結ビットに対してはバンクチャリング処理を実行しない。すなわち、この終結ビットは、バンクチャリング部に入力されることなく、多重化部47へ送られる。

【0053】図5において、選択情報「0」は、入力データエレメントを選択しないことを表し、選択情報「1」は、入力データエレメントを選択することを表す。例えば、図5に示す選択情報は、入力データ系列か

ら、第2番目、第4番目、第5番目、...、N番目のデータエレメントを選択する旨を規定している。すなわち、このバンクチャリングテーブルを用いてバンクチャリング処理が実行されると、パリティデータ系列 $Y_{1k} = Y_{11}, Y_{12}, Y_{13}, Y_{14}, Y_{15}, \dots$ が順番に入力された場合には、 $Y_{12}, Y_{14}, Y_{15}, \dots$ が選択されることになる。そして、この選択されたデータエレメントは、パリティデータ系列 $Z_{1k}$ として多重化部47へ出力される。

【0054】第2のバンクチャリング部46は、基本的に第1のバンクチャリング部45と同じである。また、第2のバンクチャリング部46に設けられるバンクチャリングテーブルも、基本的には第1のバンクチャリング部45に設けられるバンクチャリングテーブルと同じである。ただし、これら2つのテーブルに設定される選択情報は、必ずしも互いに同じである必要はない。

【0055】なお、図4に示したCPU52およびメモリ53は、第1のバンクチャリング部45および第2のバンクチャリング部46により共有可能である。さらに、選択情報として1つのバンクチャリングパターンを用意しておき、第1のバンクチャリング部45および第2のバンクチャリング部46がその選択情報を共用するようにしてもよい。

【0056】バンクチャリングテーブルは、メモリ53内のRAM領域に格納される。したがって、必要に応じて選択情報を変更することができる。このことにより、所望の符号化率を得ることができる。また、ソースデータのデータ長に応じて、或いは畳込み部の出力系列のデータ長に応じて選択情報のデータ長を変更することも可能である。

【0057】次に、バンクチャリングテーブルの作成方法（即ち、選択情報の作成方法）を説明する。以下では、ソースデータ $u$ のデータ長を $N$ ビット、出力系列 $C$ のデータ長を $M$ ビットにする旨が要求されているものとする。この場合、符号化率 $R = N/M$ が要求される。なお、第1の畳込み部43および第2の畳込み部44によりそれぞれ生成される終結ビットは、そのデータ長がソースデータ $u$ のデータ長と比べて十分に短いので、以下の説明では無視するものとする。

【0058】ソースデータ $u$ のデータ長が「 $N$ 」である場合は、データ系列 $X_k$ 、第1の畳込み部43により生成されるパリティデータ系列 $Y_{1k}$ 、および第2の畳込み部44により生成されるパリティデータ系列 $Y_{2k}$ のデータ長は、それぞれ「 $N$ 」になる。したがって、出力系列 $C$ のデータ長を $M$ ビットにするためには、第1のバンクチャリング部45および第2のバンクチャリング部46によりそれぞれ生成されるパリティデータ系列 $Z_{1k}$ および $Z_{2k}$ の各データ長をそれぞれ「 $K_1$ 」および「 $K_2$ 」とした場合、以下の式を満たす必要がある。

$$【0059】N + K_1 + K_2 = M$$

ここで、 $K_1 = K_2 = K$ とすると、下式が得られる。

$$K = (M - N) / 2$$

（ただし、 $M > N$ 、 $N > K$ ）

すなわち、この場合、第1のバンクチャリング部45は、 $N$ 個のデータエレメントから構成されるパリティデータ系列 $Y_{1k}$ から $K$ 個のデータエレメントを選択してパリティデータ系列 $Z_{1k}$ として出力することになる。同様に、第2のバンクチャリング部46は、 $N$ 個のデータエレメントから構成されるパリティデータ系列 $Y_{2k}$ から $K$ 個のデータエレメントを選択してパリティデータ系列 $Z_{2k}$ として出力することになる。

【0060】 $N$ 個のデータエレメントから $K$ 個を選択する際には、バンクチャリングテーブルが利用される。バンクチャリングテーブルに格納される選択情報は、上述したように、入力系列の各データエレメントを選択するか否かを表すので、 $K$ 個のデータエレメントを選択するには、 $N$ ビットの選択情報の中の $K$ ビットに「1：選択」割り当て、他のビットに「0：非選択」を割り当てればよい。 $N$ ビットの中の $K$ ビットに対して「1」を割り当てる方法の具体例を以下に示す。

【0061】(1) 複数のシード系列 $k/n$ を作成する。

「 $k/n$ 」は、 $k$ 個の「1」が均等に割り当てられた $n$ ビットの系列である。なお、 $k = 1, 2, 3, \dots, n = 1, 2, 3, \dots$ である。また、 $n > k$ である。シード系列は、例えば、「 $n$ 」の最大値を10、「 $k$ 」の最大値を9として作成する。シード系列の一部を示す。なお、各シード系列の先頭ビットには、「0」を割り当てる。

$$k/n = 2/7 : (0001001)$$

$$1/3 : (001)$$

$$3/8 : (00100101)$$

$$2/5 : (00101)$$

$$3/7 : (0010101)$$

$$4/9 : (001010101)$$

$$5/9 : (010101011)$$

$$1/2 : (01)$$

$$4/7 : (0110101)$$

$$3/5 : (01101)$$

$$5/8 : (01110101)$$

$$3/4 : (0111)$$

$$4/5 : (01111)$$

$$5/6 : (011111)$$

(2) 最適なシード系列を選択する。具体的には、「 $K/N \geq k/n$ 」という条件の下で、下式により得られる値 $r$ が最小となる「 $k/n$ 」をさがす。

$$【0062】$$

$$【数2】$$

$$r = \min \left[ \frac{K}{N} - \frac{k}{n} \right]$$

ただし、 $\frac{K}{N} \geq \frac{k}{n}$

【0063】たとえば、ソースデータuのデータ長Nが「300」であり、バンクチャリング処理によって300個のデータエレメントから155個を選択する場合に、 $K/N = 155/300$ を代入すれば、「 $k/n$ 」として「 $1/2$ 」が得られる。また、この場合、 $r = 0.01666$ が得られる。

【0064】(3) 上記(2)で選択したシード系列を用いてバンクチャリングテーブルに書き込むべき選択情報のベースパターンを作成する。具体的には、選択したシード系列を繰り返すことにより、データ長=Nのベースパターンを作成する。例えば、上述の例のように、 $k/n = 1/2$ のシード系列が選択された場合には、シード系列(01)を繰り返すことにより、300ビットのベースパターンを得る。

【0065】(4) ベースパターンを修正することにより、選択情報を得る。具体的には、まず、 $A = r \cdot N$ を算出する。そして、上記(3)において作成したベースパターンにおいて、A個の「0」を均等に選択し、それらを「1」に置き換える。なお、ベースパターンの先頭ビットは、選択しない。たとえば、上述の例の場合には、 $A = 0.01666 \times 300 = 5$ が得られるので、ベースパターン(01010101...0101)において、5個の「0」が「1」に置き換えられる。

【0066】上記処理により得られたパターンは、選択情報(バンクチャリングパターン情報)としてバンクチャリングテーブルに格納される。第1のバンクチャリング部4.5および第2のバンクチャリング部4.6にそれぞれ設けられるバンクチャリングテーブルは、一実施例としては、互いに同じものを用いるが、それら2つのテーブルは必ずしも互いに同じである必要はない。ただし、それら2つのテーブルに格納される各選択情報に含まれる「1」の数は、互いに同じであるか、あるいは互いに近接していることが望ましい。各選択情報に含まれる「1」の数の差が互い大きく異なると、復号特性が悪くなる恐れがある。

【0067】選択情報の先頭ビットを「0」にする理由は以下の通りである。選択情報の先頭ビットは、第1の畳込み部4.3により生成されるパリティデータ系列Y1k(または、第2の畳込み部4.3により生成されるパリティデータ系列Y2k)の先頭データエレメントを選択するか否かを表す。パリティデータ系列Y1kの先頭データエレメントは、第1の畳込み部4.3において、ソースデータuの先頭データエレメントと図3に示したメモリMに格納されている初期値との加算演算により生成される。

ところが、この初期値は、一般に、「オール0」であるので、パリティデータ系列Y1kの先頭データエレメントは、ソースデータuの先頭データエレメントそのものである。すなわち、「畳込み」による効果は得られない。したがって、選択情報の先頭ビットに「1」を割り当てることにより、そのパリティデータ系列Y1kの先頭データエレメントを選択して受信装置へ送ったとしても、復号処理において誤り訂正能力を向上させることには寄与しない。

【0068】このため、本実施形態では、選択情報の中の「1」を先頭以外の他のデータエレメントを選択するために割り当てることにより、復号処理における誤り訂正能力の向上を図っている。

【0069】次に、バンクチャリングテーブルを用いたバンクチャリング処理を説明する。第1のバンクチャリング部4.5は、パリティデータ系列Y1kのデータエレメントを受け取るごとに、バンクチャリングテーブルを参照して、そのデータエレメントを選択するか否かを判断する。選択したデータエレメントは、パリティデータ系列Z1kとして多重化部4.7へ送出される。一方、選択されなかったデータエレメントは、多重化部4.7へ送出されることなく廃棄される。この処理は、第2のバンクチャリング部4.6においても同様である。

【0070】図6は、バンクチャリング処理のフローチャートである。この処理は、畳込み部により生成されるパリティデータ系列Ykのデータエレメントがラッチ回路5.1に書き込まれるごとに実行される。なお、パリティデータ系列Ykは、パリティデータ系列Y1kまたはY2kを表す。即ち、このフローチャートの処理が第1のバンクチャリング部4.5の動作を表す場合には、 $Yk = Y1k$ であり、このフローチャートの処理が第2のバンクチャリング部4.6の動作を表す場合には、 $Yk = Y2k$ である。

【0071】ステップS1では、ラッチ回路5.1からデータエレメントを取得する。ステップS2では、ラッチ回路5.1に書き込まれたデータエレメントがパリティデータ系列Yk内の第何番目のデータエレメントであるのかを計数するためのカウンタをインクリメントする。このカウンタ値kは、データエレメントの位置情報あるいはシーケンス番号に対応する。なお、このカウンタは、1セットのソースデータに対する処理が終了することによりリセットされる。

【0072】ステップS3では、上記カウンタのカウンタ値kを用いて、図5に示すバンクチャリングテーブルを参照する。このことにより、ラッチ回路5.1に書き込まれたデータエレメントに関する選択情報P(k)が得られる。ステップS4では、ステップS3で得た選択情報P(k)が「1」であるか「0」であるかを調べる。選択情報P(k) = 1であれば、ステップS5において、ラッチ回路5.1に書き込まれたデータエレメントを出力ホー

ト54を介して多重化部47へ送出する。このとき、バンクチャリングテーブルを参照する際に用いたカウント値 $k$ も多重化部47へ送る。一方、選択情報 $P(k) = 0$ であれば、ステップS6において、ラッチ回路51に書き込まれたデータエレメントを廃棄する。

【0073】ステップS7では、カウント値 $k$ が「N」に達したか否かを調べる。カウント値 $k$ が「N」に達していれば、1セットのソースデータに対する処理が終了したものとみなし、ステップS8において上記カウンタをリセットする。

【0074】このように、第1のバンクチャリング部4

		$y_1(3)$	$y_1(4)$		$y_1(6)$			$y_1(9)$
		$y_2(3)$	$y_2(4)$		$y_2(6)$			$y_2(9)$

【0077】この出力は、入力されるソースデータ $u$ が9ビットであり、また、第1のバンクチャリング部45および第2のバンクチャリング部46におけるバンクチャリングパターン $P$ が共に(001101001)であ

【0078】図7は、多重化部47のブロック図である。多重化部47は、データ系列 $X_k$ を保持するためのバッファ61、第1のバンクチャリング部45により生成されたパリティデータ系列 $Z_{1k}$ を保持するためのメモリ62、第2のバンクチャリング部46により生成されたパリティデータ系列 $Z_{2k}$ を保持するためのメモリ63およびバッファ61、およびメモリ62、63からデータエレメントを読み出して出力する読出し制御部64を備える。

【0079】データ系列 $X_k$ のデータエレメントは、バッファ61に順番に書き込まれる。パリティデータ系列 $Z_{1k}$ は、第1のバンクチャリング部45により選択されたデータエレメントである。これらのデータエレメントは、シーケンス番号に対応付けられてメモリ62に書き込まれる。各データエレメントに対応するシーケンス番号は、たとえば、図6を参照しながら説明したカウンタのカウント値 $k$ により指示される。なお、メモリ62は、シーケンス番号毎にそれぞれデータエレメントが書き込まれているか否かを表す「有効/無効表示」が

【0080】読出し制御部64は、予め決められた所定の間隔ごとに、バッファ61、およびメモリ62、63の中のいずれか1つから1つのデータエレメントを読み出して出力する。具体的には、下記の手順(1)～(4)を繰り返して実行することによりデータエレメントを読み出す。

(1) バッファ61から指定されているシーケンス番号のデータエレメントを読み出す。

5および第2のバンクチャリング部46は、入力される $N$ ビットのパリティデータ系列 $Y_k$ から $K$ ビットを選択して出力する。なお、この選択処理は、上記ステップS1～S8を記述したプログラムをCPU52に実行させることにより実現される。

【0075】第1のバンクチャリング部45および第2のバンクチャリング部46による出力の例を表2に示す。

【0076】

10 表2]

(2) 上記指定されているシーケンス番号のデータエレメントがメモリ62に格納されている場合には、それを読み出す。

(3) 上記指定されているシーケンス番号のデータエレメントがメモリ63に格納されている場合には、それを読み出す。

(4) 次のシーケンス番号を指定する。

【0081】バッファ61、メモリ62、63が図7に示す状態であった場合に、上記手順(1)～(4)を繰り返して実行すると、出力系列 $C$ は、以下のようになる。出力系列 $C = (X_1, X_2, X_3, Y_{13}, Y_{23}, X_4, Y_{14}, Y_{24}, X_5, \dots)$  このように、図3に示した誤り訂正符号化装置40は、バンクチャリング部に格納されている選択情報(バンクチャリングパターン)を用いて、誤り訂正のために付加されるパリティデータのデータ量を変えることができる。すなわち、選択情報の設定に従って所望の符号化率 $R$ が得られる。

【0082】次に、上記誤り訂正符号化装置40によって符号化されたデータ系列を復号する復号装置について簡単に説明しておく。復号処理としては、様々な方法が知られているが、基本的には、符号化処理を逆の手順で実行することによりデータ系列を復号する。

【0083】図8は、復号装置のブロック図である。ここでは、誤り訂正符号化装置40の第1のバンクチャリング部45および第2のバンクチャリング部46において互いに同じ選択情報を用いてパリティデータ $Y_{1k}$ および $Y_{2k}$ に対してバンクチャリング処理が実行されたものとする。また、この復号装置は、特に図示していないが、誤り訂正符号化装置40において多重化されたデータ系列 $X$ とパリティデータ系列 $Z$ とを分離する機能を備えている。

【0084】シリアル/パラレル変換器71は、受信したパリティデータ系列 $Z$ をパリティデータ系列 $Z_{1k}$ および $Z_{2k}$ に分離する。パリティデータ系列 $Z_{1k}$ および $Z_{2k}$

は、それぞれ誤り訂正符号化装置 40 の設けられている第 1 のバンクチャリング部 45 および第 2 のバンクチャリング部 46 により生成された系列である。

【0085】第 1 のデバンクチャリング部 72 および第 2 のデバンクチャリング部 73 は、それぞれ誤り訂正符号化装置 40 が有するバンクチャリングテーブルと同じテーブルを備え、それぞれパリティデータ系列 Z1k および Z2k に対して「デバンクチャリング処理」を実行する。

【0086】図 9 を参照しながら、デバンクチャリング処理の一例を説明する。ここでは、パリティデータ系列 Z1k = (Z11, Z12, Z13, Z14, Z15) が入力されたものとする。また、バンクチャリングテーブルには図 10 に示す選択情報が格納されていたものとする。なお、以下では、第 1 のデバンクチャリング部 72 の処理を説明するが、第 2 のデバンクチャリング部 73 の処理も同じである。

【0087】第 1 のデバンクチャリング部 72 は、パリティデータ系列 Z1k を受け取ると、まず、バンクチャリングテーブルのシーケンスナンバ = 「1」に対応する選択情報を調べる。この場合、選択情報 = 「0」なので、第 1 のデバンクチャリング部 72 は、「0」を出力する。つづいて、バンクチャリングテーブルのシーケンスナンバ = 「2」に対応する選択情報を調べる。この場合、選択情報 = 「1」なので、第 1 のデバンクチャリング部 72 は、パリティデータ系列 Z1k の先頭データエレメントである「Z11」を出力する。以下、同様に、第 1 のデバンクチャリング部 72 は、選択情報 = 「0」の場合には「0」を出力し、選択情報 = 「1」の場合には、パリティデータ系列 Z1k のデータエレメントを 1 つずつ 30 順番に出力していく。この結果、第 1 のデバンクチャリング部 72 は、下記のデータ系列を出力する。

【0088】出力系列: (0, Z11, 0, Z12, 0, Z13, 0, Z14, Z15)

この出力系列は、パリティデータ系列 Y1k として、第 1 の復号器 74 に与えられる。同様に、第 2 のデバンクチャリング部 73 は、パリティデータ系列 Y2k を生成して第 2 の復号器 75 に与える。

【0089】図 10 は、デバンクチャリング処理のフローチャートである。ここでは、入力データ系列 Z に対してデータ系列 Y を生成する場合を説明する。データ系列 Z およびデータ系列 Y のデータエレメントを、それぞれ「Zi」「Yk」と表す。

【0090】ステップ S11 では、「k」を用いてバンクチャリングテーブルをサーチし、対応する選択情報を取得する。ステップ S12 では、ステップ S11 で取得した選択情報が「1」であるか「0」であるかを調べる。取得した選択情報が「1」であれば、ステップ S13 において、データ系列 Yk のデータエレメントとしてデータ系列 Zi のデータエレメントを 1 つ出力する。そ 50

して、ステップ S14 において、「i」をインクリメントする。一方、取得した選択情報が「0」であったときには、ステップ S15 において、データ系列 Yk のデータエレメントとして「0」を出力する。

【0091】続いて、ステップ S16 では、「k」をインクリメントする。そして、ステップ S17 において、「k」が「N」に達したか否かを調べる。「N」は、ソースデータのデータ長である。「k」が「N」に達していなければ、ステップ S11 に戻り、「k」が「N」に達していれば、「k」及び「i」をリセットする。

【0092】図 8 に戻る。第 1 のデバンクチャリング部 72 により生成されたパリティデータ系列 Y1k は、第 1 の復号器 74 に与えられる。同様に、第 2 のデバンクチャリング部 73 により生成されたパリティデータ系列 Y2k は、第 2 の復号器 75 に与えられる。第 1 の復号器 74 は、パリティデータ系列 Y1k を利用しながら受信したデータ系列 Xk を復号する。また、第 2 の復号器 75 は、パリティデータ系列 Y2k を利用しながら第 1 の復号器 74 の出力を復号する。

【0093】第 2 の復号器 75 の出力は、判定部 76 において予め決められた閾値と比較される。そして、デインターリーバ 77 においてその比較結果に対してデインターリービング処理（誤り訂正符号化装置 40 におけるランダム化処理の逆の処理）が実行され、その結果が復号データとして出力される。

【0094】なお、パリティデータ系列を生成する処理を除く復号処理は、既知の技術を利用して実現可能である。たとえば、米国特許 5,446,747 に記載されている。したがって、ここでは、復号処理についての詳しい説明を省略する。

【0095】復号精度を向上させるためには、図 11 に示すように、上記構成の復号装置を直列に接続すればよい。この場合、図 8 に示した復号装置が 1 つの復号モジュールに相当する。各復号モジュールは、受信データ系列（復号すべきデータ系列 X およびパリティデータ系列）、および前段の復号モジュールにおけるデータ系列 X の予測値（系列 T）を受け取り、復号データ S を生成すると共に、新たにデータ系列 X を予測する。この新たに予測されたデータ系列 X は、次段の復号モジュールに渡される。

【0096】上記構成においては、直列に接続される復号モジュールの数を増やすことにより、復号精度は向上する。たとえば、復号モジュール 70-1 から出力される復号データ S よりも復号モジュール 70-4 から出力される復号データ S の方が復号精度は高くなる。なお、この構成の動作は、米国特許 5,446,747 に記載されている。

【0097】図 11 に構成において、図 8 に示したシリアル/パラレル変換部 71、第 1 のデバンクチャリング部 72、および第 2 のデバンクチャリング部 73 は、第

1段目の復号モジュール70-1に対して設ければよい。

【0098】次に、本発明の他の実施形態の誤り訂正符号化装置について説明する。従来の誤り訂正符号化装置は、上述したように、一般に、符号化率が固定的に与えられていた。たとえば、図20に示した構成では、符号化率 $R=1/2$ であり、図21に示した構成では、符号化率 $=1/3$ であった。以下に説明する誤り訂正符号化装置では、任意の符号化率が得られる。特に、 $1/3$ よりも小さい任意の符号化率が得られる。

【0099】図12は、本実施形態の誤り訂正符号化装置40と従来の装置との出力の差異を示す図である。ここでは、従来の装置として、図21に示した装置を採用上げる。従来の装置では、図23を参照しながら説明したように、たとえば、ソースデータのデータ長が666ビット、要求されている出力データ長が1500ビットである場合には、符号化データに対して168ビットのダミーデータが付与される。この場合、エラーを訂正するために使用されるパリティデータは、666ビットとなる。

【0100】一方、誤り訂正符号化装置40を使用した場合には、図p1に示すように、666ビットのパリティデータ系列Y1KおよびY2Kから、それぞれ417ビットのパリティデータ系列Z1KおよびZ2Kが生成される。この結果、エラーを訂正するために使用されるパリティデータは、834ビットとなる。すなわち、従来の装置と比較して、エラーを訂正するために使用されるデータ量が多くなる。したがって、本実施形態においては、復号能力が高くなる。

【0101】図13は、本発明の他の形態の誤り訂正符号化装置80の構成図である。図13において、インターリバ42、第1の畳込み部43、第2の畳込み部44、および多重化部47は、図3に示したものと同一である。また、図13では、入力I/F部41を省略している。

【0102】誤り訂正符号化装置80の特徴は、ビット重複部81を設けたことである。ビット重複部81は、所望の符号化率を得るために、ソースデータuの中の所定数のデータエレメントを重複させる。

【0103】ビット重複部81の動作を説明する。以下では、ソースデータuのデータ長がNビット、出力データ系列のデータ長がMビットであるとする。また、 $M>3N$ とする。すなわち、符号化率として $1/3$ よりも小さい値が要求されている場合を想定する。

【0104】ビット重複部81においてソースデータuの中のrビットを重複させることによりデータ系列Xkを得るものとする。データ系列Xk、パリティデータ系列Y1k、およびパリティデータ系列Y2kの各データ長は、それぞれ「N+r」になる。したがって、出力データ系列のデータ長をMビットにするためには、ビット重

複部81において重複させるべきビット数は下式により得られる。

$$【0105】(N+r) \times 3 = M$$

$$\therefore r = M/3 - N$$

たとえば、ソースデータuのデータ長を250ビット、要求される出力系列のデータ長を900ビットとすると、上式に $N=250$ 、 $M=900$ を代入することにより、 $r=50$ が得られる。

【0106】ビット重複部81は、望ましくは、「拘束長+1」ビットごとにソースデータuのデータエレメントを重複させる。ここで、拘束長は、畳込み処理においてメモリに保持されるデータのビット数である。たとえば、図13に示す構成においては、拘束長=2なので、3ビットごとにソースデータuのデータエレメントが重複される。

【0107】このように、所定のデータエレメントを重複させたデータ系列を符号化して伝送すると、符号技術において既知であるように、データエレメントが重複している部分の後続のデータエレメントに対する復号処理の精度が向上する。

【0108】図14は、ビット重複部81の動作例を示す図である。ここでは、ソースデータuのデータ長が7ビット、拘束長=2、要求される出力系列のデータ長が27ビットの場合を示している。この場合、2個のデータエレメントが重複される。また、3ビット毎にデータエレメントが重複される。この処理により、誤り訂正符号化装置80の符号化率は、「 $7/27$ 」になる。

【0109】図15は、ビット重複部81の動作を説明するフローチャートである。ここでは、ソースデータu ( $u_0, u_1, u_2, u_3, \dots, u_i, \dots$ ) が入力されたものとする。また、重複すべきデータエレメントの数を「r」とする。さらに、xビット毎にデータエレメントを重複する。

【0110】ステップS21では、ソースデータuのデータエレメント $u_i$ を取得する。以下では、「i」をシーケンスナンバと呼ぶことにする。ステップS22では、ビット重複数jが重複すべきデータエレメントの数である「r」に達しているか否かを調べる。ビット重複数jは、当該ソースデータuにおいて既にビット重複を実行した回数を表す。 $j>r$ であれば、すでに必要な数のビット重複を終了しているものとみなし、ステップS23において、取得したデータエレメント $u_i$ をそのまま出力する。一方、 $j \leq r$ であれば、更にビット重複を行う必要があるとみなし、ステップS24に進む。

【0111】ステップS24では、シーケンスナンバiが「x」の整数倍であるか否かを調べる。シーケンスナンバiが「x」の整数倍でなければ、ビット重複を行わないので、ステップS23へ進む。一方、シーケンスナンバiが「x」の整数倍であったときには、ビット重複を行う。すなわち、ステップS25およびS26におい

でそれぞれデータエレメント $u_i$ を出力する。このことにより、データエレメント $u_i$ は重複される。続いて、ステップS27において、ビット重複数 $j$ をインクリメントする。

【0112】ステップS28では、シーケンスナンバ $i$ が「N」に達したか否かを調べる。シーケンスナンバ $i$ が「N」に達していない場合には、ステップS29においてそのシーケンスナンバ $i$ をインクリメントした後、ステップS21に戻って次のデータエレメントを取得する。一方、シーケンスナンバ $i$ が「N」に達している場合には、当該ソースデータのすべてのデータエレメントについてステップS21～S29の処理が実行されたものとみなし、ステップS30において「 $i$ 」および「 $j$ 」をリセットした後に処理を終了する。

【0113】このように、図13に示した誤り訂正符号化装置80は、所望の符号化率を得るために、ソースデータの中の所定数のデータエレメントを重複させる。換言すれば、ソースデータの中の所定数のデータエレメントを重複させることにより、所望の符号化率が得られる。また、重複ビットは、復号処理において利用されるので、伝送路の誤り率を小さくすることができる。

【0114】なお、誤り訂正符号化装置80によって符号化されたデータ系列を復号する復号装置は、通常の復号処理を実行した後に、ビット重複部81による処理を逆の手順で実行する機能を設ければよい。

【0115】図16は、本発明の更に他の形態の誤り訂正符号化装置90の構成図である。図16において、インターリーバ42、第1の畳込み部43、第2の畳込み部44、および多重化部47は、図3に示したものと同一である。

【0116】誤り訂正符号化装置90の特徴は、ダミービット挿入部91を設けたことである。ダミービット挿入部91は、所望の符号化率を得るために、ソースデータ $u$ に対して所定数のダミービットを挿入する。

【0117】ダミービット挿入部91の動作を説明する。以下では、ソースデータ $u$ のデータ長が $N$ ビット、出力データ系列のデータ長が $M$ ビットであるとする。また、 $M > 3N$ とする。すなわち、符号化率として $1/3$ よりも小さい値が要求されている場合を想定する。

【0118】ダミービット挿入部91においてソースデータ $u$ に対して $r$ ビットのダミービットを挿入することによりデータ系列 $X_k$ を得る場合、データ系列 $X_k$ 、パリティデータ系列 $Y_{1k}$ 、及びパリティデータ系列 $Y_{2k}$ の各データ長は、それぞれ「 $N+r$ 」になる。従って、出力データ系列のデータ長を $M$ ビットにするためには、ダミービット挿入部91において挿入すべきビット数は下式により得られる。

$$【0119】(N+r) \times 3 = M$$

$$\therefore r = M/3 - N$$

ダミービット挿入部91は、望ましくは、拘束長と同じ

長さのダミービットを挿入する。拘束長は、上述したように、畳込み処理においてメモリに保持されるデータのビット数である。したがって、図13に示す構成においては、ダミービットは、2ビットを1単位としてソースデータ $u$ に挿入される。

【0120】ダミービットとしては、「1」または「0」を用いる。ここで、もしダミービットとして「1」を用いるとすると、拘束長=2の場合には、ダミーデータとして「11」が挿入される。例えば、ソースデータ $u$ のデータ長を250ビット、要求される出力系列のデータ長を900ビットとすると、 $r=50$ が得られる。すなわち、ソースデータ $u$ に対して50ビットのダミービットを挿入することが要求される。ここで、拘束長=2とすると、ソースデータ $u$ に対して「11」が25カ所に挿入されることになる。挿入位置は、均等に分散させることが望ましい。

【0121】このように、ダミーデータとして「1」が挿入されたデータ系列を符号化して伝送すると、符号技術において既知であるように、そのダミービットの後続のデータエレメントに対する復号処理の精度が向上する。

【0122】なお、図22および図23を参照しながら説明したように、従来の誤り訂正符号化装置においてもしばしばダミーデータが使用されていた。しかしながら、従来の方法では、符号化されたデータ系列に対してダミーデータを付加していたのに対し、誤り訂正符号化装置90では、ソースデータにダミービットを挿入し、そのダミービットが挿入されたソースデータを符号化する構成である。即ち、従来の方法においては、ダミーデータは、「無意味なデータ」であったが、誤り訂正符号化装置80においては、ダミービットは、復号処理において事前確率尤度として利用されるので、有用なデータである。

【0123】図17は、ダミービット挿入部91の動作例を示す図である。ここでは、ソースデータ $u$ のデータ長が7ビット、拘束長=2、要求される出力系列のデータ長が27ビットの場合を示している。この場合、ソースデータ $u$ に対して2ビットのダミービットを挿入することにより、符号化率=7/27を実現している。

【0124】このように、図16に示した誤り訂正符号化装置90は、所望の符号化率を得るために、ソースデータの中の所定数のダミービットを挿入する。換言すれば、ソースデータの中の所定数のダミービットを挿入することにより、所望の符号化率が得られる。また、挿入されたダミービットは、復号処理において利用されるので、伝送路の誤り率を小さくすることができる。

【0125】なお、誤り訂正符号化装置90によって符号化されたデータ系列を復号する復号装置は、通常の復号処理を実行した後に、ダミービットを除去する機能を設ければよい。

【0126】上記図3、図13、および図16に示した誤り訂正符号化装置は、それぞれ互いに並列に接続された2つの畳込み部を備える構成であったが、本発明は、この構成に限定されるものではない。すなわち、本発明は、互いに並列に接続された複数の畳込み部を有する装置に適用可能である。

【0127】図18は、 $m$ 個の畳込み部を備える誤り訂正符号化装置100のブロック図である。畳込み部101-1~101- $m$ は、それぞれソースデータに対して畳み処理を実行する。畳込み部101-2~101- $m$ に対して互いに異なるインターリーブが設けられている。このことにより、畳込み部101-1~101- $m$ に対して互いに異なる系列が与えられる。

【0128】バンクチャリング部102は、畳込み部101-1~101- $m$ からそれぞれ出力されるパリティデータ系列 $Y_{1k} \sim Y_{mk}$ の中から所定数のデータエレメントを選択して出力する。たとえば、ソースデータ $u$ のデータ長を $N$ ビット、出力系列 $C$ のデータ長を $M$ ビットである場合、すなわち符号化率 $=N/M$ である場合、バンクチャリング部102は、以下のようにしてデータエレメントを選択する。ここで、各畳込み部101-1~101- $m$ は、 $N$ ビットの系列が与えられたときに $N$ ビットのパリティデータを出力するとする。

【0129】バンクチャリング部102によりパリティデータ系列 $Y_{1k} \sim Y_{mk}$ からそれぞれ $K_1 \sim K_m$ 個のデータエレメントを選択するものとすると、下式が得られる。

$$N + K_1 + K_2 + K_3 + \dots + K_m = M$$

ここで、 $K_1 = K_2 = K_3 = \dots = K_m = K$ とすると、下式が得られる。

$$\begin{aligned} \text{【0130】} & K = (M - N) / m \\ \therefore \text{符号化率 } R &= N / M = (M - m \cdot K) / M \\ & \text{〔ただし、} M > N, N > K \text{〕} \end{aligned}$$

このように、誤り訂正符号化装置の符号化率 $R$ は、並列に設けられる畳込み部の数、および $N$ ビットの系列から選択すべきデータエレメントの数に応じて決めることができる。

【0131】なお、上記実施例では、図3、図13、図16に示す誤り訂正符号化装置が互いに独立しているものとして説明したが、これらの装置を任意に組み合わせることができる。たとえば、図3に示した誤り訂正符号化装置40の入力部に、図13に示したビット重複部81あるいは図16に示したダミービット挿入部91をもうけるようにしてもよい。

【0132】また、上記実施例の誤り訂正符号化装置は、組織符号を採用すると共に、畳込み処理を実行する構成を採用していた。しかしながら、本発明は、この構成に限定されるものではない。すなわち、本発明の誤り訂正符号化装置は、必ずしも組織符号に限定されるものではなく、また、必ずしも畳込み部を含む構成に限定さ

れるものでもない。

【0133】図19は、組織符号に限定されない誤り訂正符号化装置のブロック図である。誤り訂正符号化装置110は、ソース系列を符号化するための複数のエンコーダ111を有する。各エンコーダ111は、畳込み符号であってもよいし、他のブロック符号（例えば、ハミング符号、BCH符号など）であってもよい。また、各エンコーダ111に与えられる系列が互いに異なるようにインターリーブ112が設けられる。バンクチャリング処理および多重化処理については、上述した実施例の構成を利用することができる。

【0134】

【発明の効果】ソースデータを符号化する誤り訂正符号化装置において、所望の符号化率（情報率）が得られる。したがって、この装置を通信システムに利用すれば、無意味なデータを伝送する必要がなくなるので、伝送効率が向上するとともに、復号特性が向上する。

【図面の簡単な説明】

【図1】本実施形態の誤り訂正符号化装置が適用される移動体通信システムの構成図である。

【図2】本実施形態の誤り訂正符号化装置が適用される記憶装置の構成図である。

【図3】本発明の一実施形態の誤り訂正符号化装置の構成図である。

【図4】バンクチャリング部のブロック図である。

【図5】バンクチャリングテーブルの一例を示す図である。

【図6】バンクチャリング処理のフローチャートである。

30 【図7】多重化部のブロック図である。

【図8】復号装置のブロック図である。

【図9】デバンクチャリング処理を説明する図である。

【図10】デバンクチャリング処理のフローチャートである。

【図11】復号精度を向上させた復号装置の一例のブロック図である。

【図12】本実施形態の誤り訂正符号化装置の出力と従来の装置の出力との差異を示す図である。

40 【図13】本発明の他の形態の誤り訂正符号化装置の構成図である。

【図14】ビット重複部の動作を説明する図である。

【図15】ビット重複部の動作を説明するフローチャートである。

【図16】本発明のさらに他の形態の誤り訂正符号化装置の構成図である。

【図17】ダミービット挿入部の動作を説明する図である。

【図18】 $m$ 個の畳込み部を備える誤り訂正符号化装置のブロック図である。

50 【図19】組織符号に限定されない誤り訂正符号化装置



のブロック図である。

【図 20】 畳込み符号を用いた既存の誤り訂正符号化装置の一例のブロック図である。

【図 21】 図 20 に示した誤り訂正符号化装置の変形例のブロック図である。

【図 22】 (a) は、図 20 に示す誤り訂正符号化装置を使用してソースデータを符号化して固定長のフレームに格納する場合の処理を説明する図である。(b) は、フレームに格納されるデータを模式的に表した図である。

【図 23】 (a) は、図 21 に示す誤り訂正符号化装置を使用してソースデータを符号化して固定長のフレームに格納する場合の処理を説明する図である。(b) は、フレームに格納されるデータを模式的に表した図である。

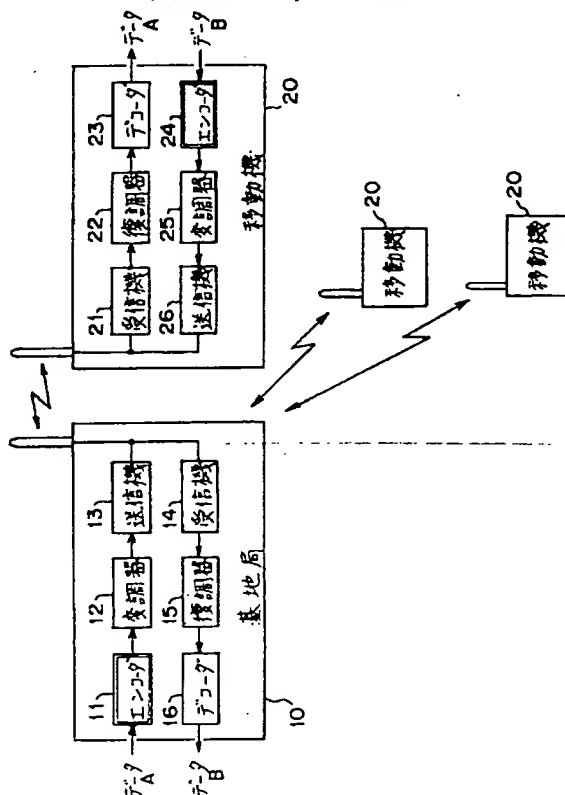
【符号の説明】

10 基地局

- 11、24 エンコーダ
- 20 移動機
- 30 記憶装置
- 31 エンコーダ
- 40 誤り訂正符号化装置
- 41 入力 I/F 部
- 42 インターリーバ
- 43 第 1 の畳込み部
- 44 第 2 の畳込み部
- 45 第 1 のバンクチャリング部
- 46 第 2 のバンクチャリング部
- 47 多重化部
- 81 ビット重複部
- 91 ダミービット挿入部

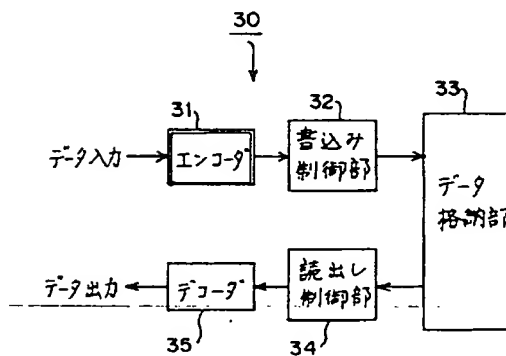
【図 1】

本実施形態の誤り訂正符号化装置が適用される  
移動体通信システムの構成図



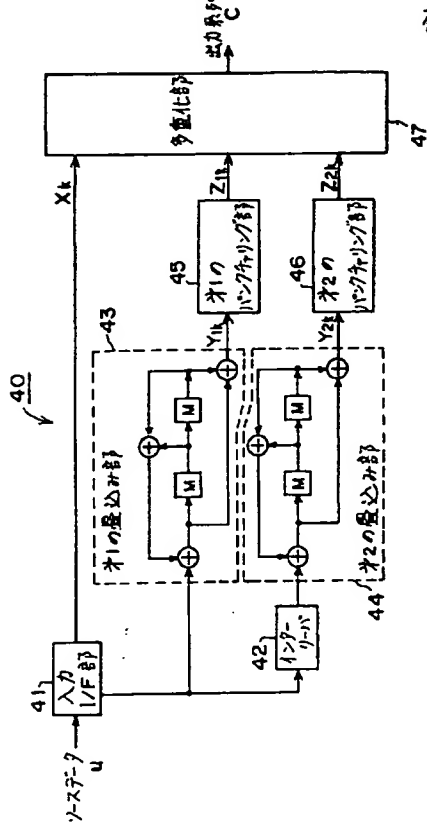
【図 2】

本実施形態の誤り訂正符号化装置が  
適用される記憶装置の構成図



【図3】

本発明の一実施形態の誤り訂正符号化装置の構成図



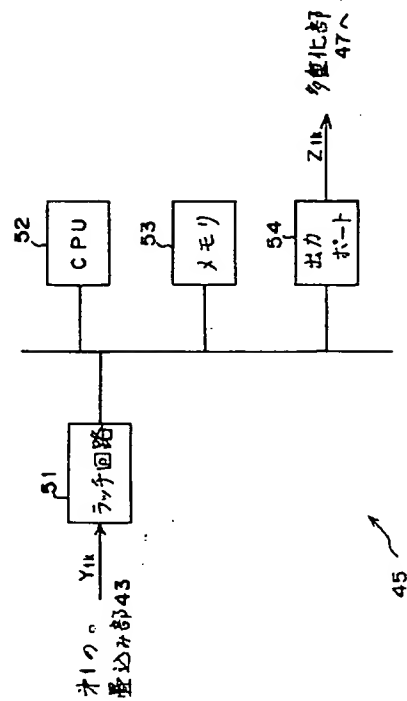
【図5】

パンクチャリングテーブルの一例を示す図

位置 (シーテンスナンバー)	1	2	3	4	5	.....	N
選択情報	0	1	0	1	1	.....	1

【図4】

パンクチャリング部のブロック図



【図9】

デパンクチャリング処理を説明する図

$$Z_k = (Z_1, Z_2, Z_3, Z_4, Z_5)$$



72 または 73

位置 (シーテンスナンバー)	1	2	3	4	5	6	7	8	9
選択情報 (パンクチャリング パターン)	0	1	0	1	0	1	0	1	1

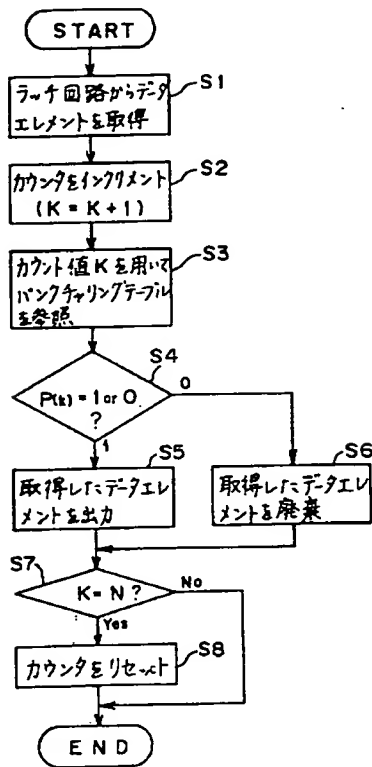
デパンクチャリング部



$$Y_k = (0, Z_1, 0, Z_2, 0, Z_3, 0, Z_4, Z_5)$$

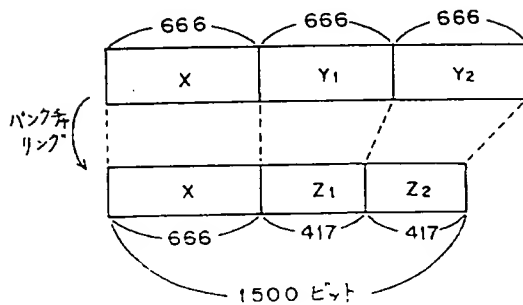
【図6】

バンクチャリング処理のフローチャート



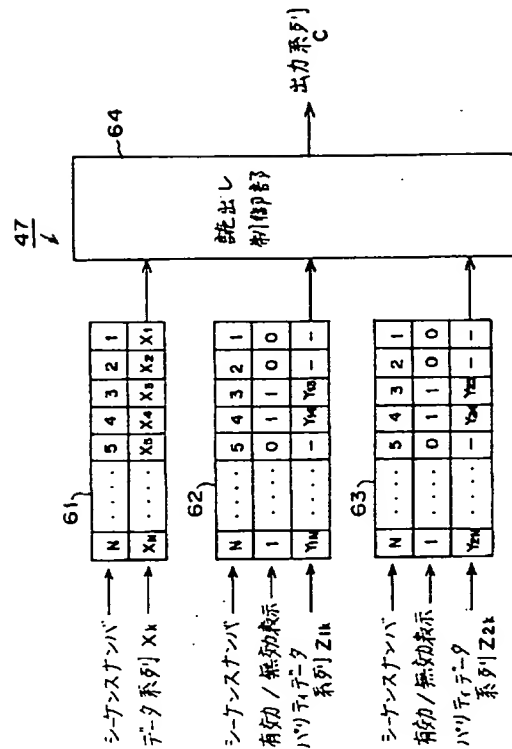
【図12】

本実施形態の誤り訂正符号化装置の  
出力と従来の装置の出力との差異を示す図



【図7】

多重化部のブロック図



【図14】

ビット重複部の動作を説明する図

$$U = U_0 U_1 U_2 U_3 U_4 U_5 U_6$$



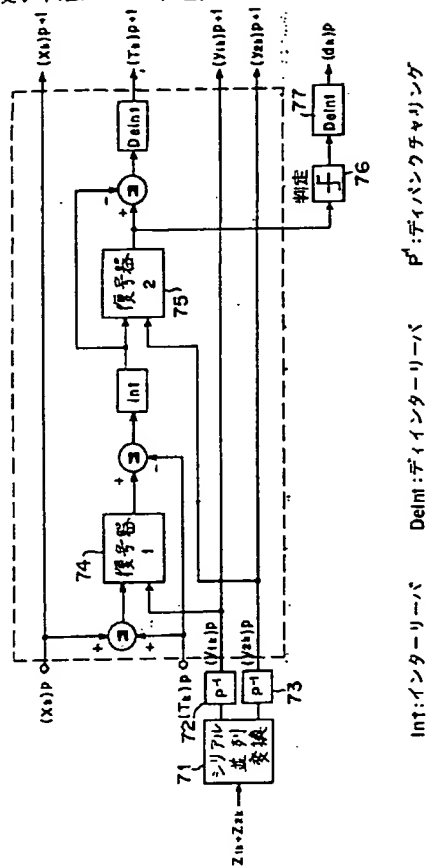
ビット重複部 81



$$\text{出力系列} = U_0 U_1 U_2 \underline{U_2} U_3 U_4 U_5 \underline{U_5} U_6$$

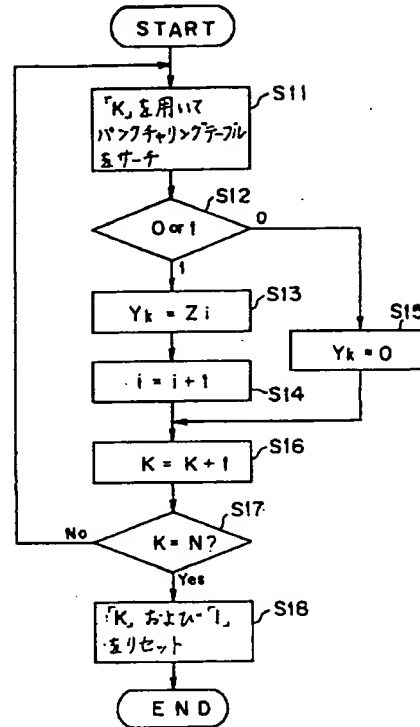
【図 8】

復号装置のブロック図



【図 10】

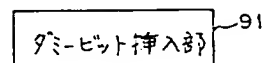
デバンクチャリング処理を説明する図



【図 17】

ダミービット挿入部の動作を説明する図

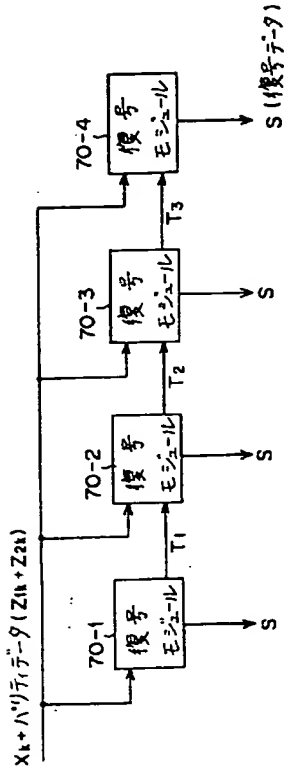
$$U = U_0 U_1 U_2 U_3 U_4 U_5 U_6$$



$$\text{出力系列} = U_0 U_1 U_2 11 U_3 U_4 U_5 U_6$$

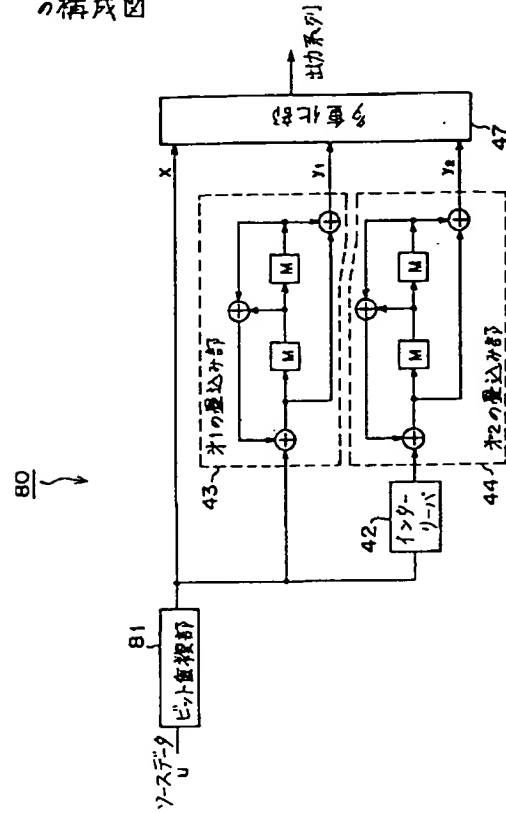
【図11】

復号精度を向上させた復号装置の一例  
のブロック図



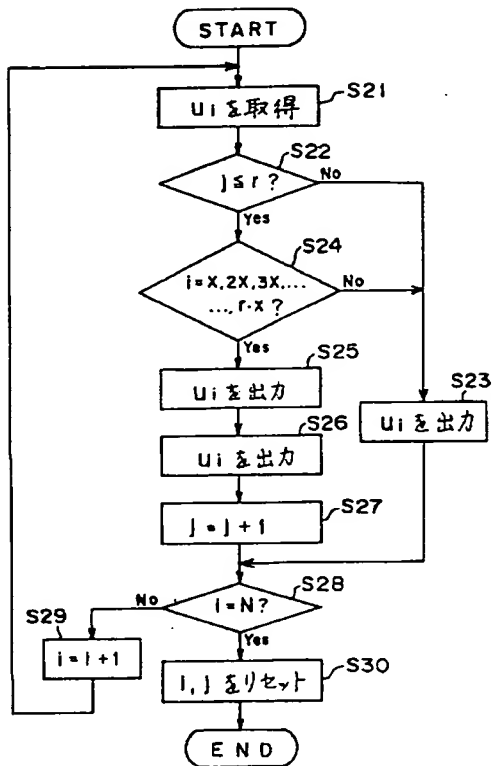
【図13】

本発明の他の形態の誤り訂正符号化装置  
の構成図



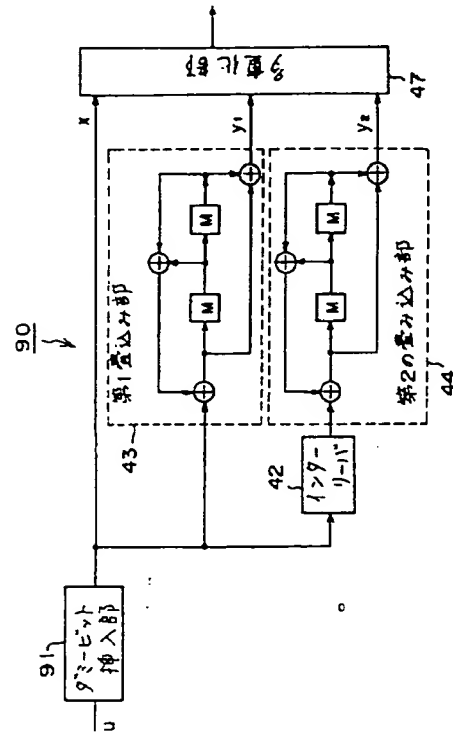
【図15】

ビット重複部の動作を説明するフローチャート



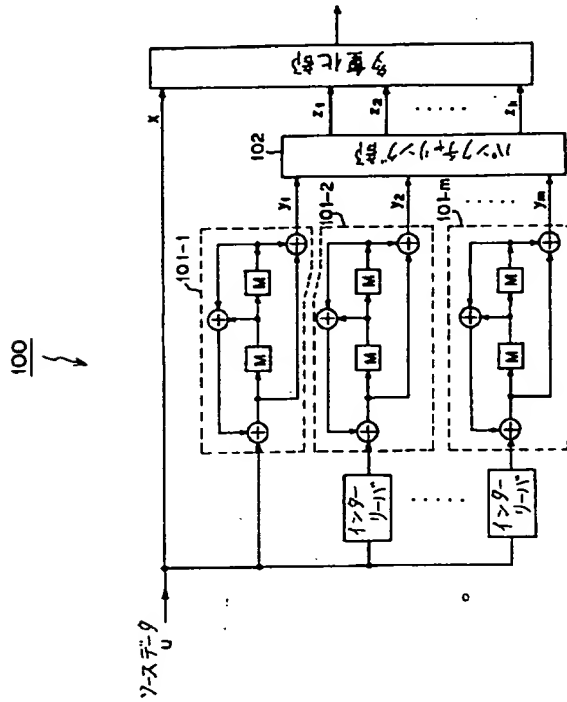
【図16】

本発明のさらに他の形態の誤り訂正  
符号化装置の構成図



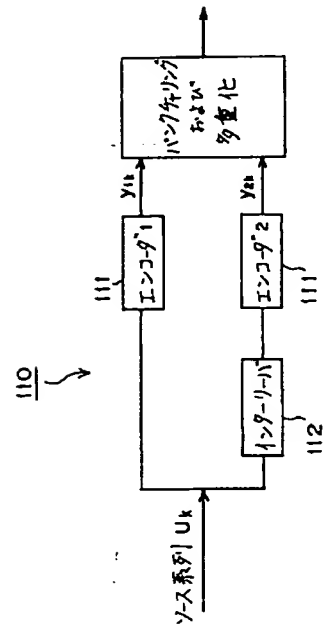
【図18】

m個の畳込み部を備える誤り訂正符号化装置のブロック図



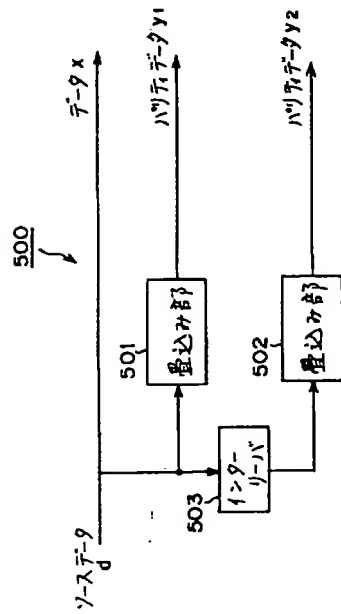
【図19】

組織符号に限定されない誤り訂正符号化装置のブロック図



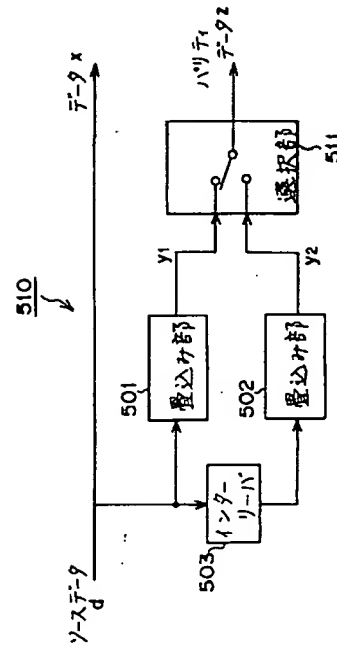
【図 20】

畳込み符号を用いた既存の誤り訂正符号化  
装置の一例のブロック図



【図 21】

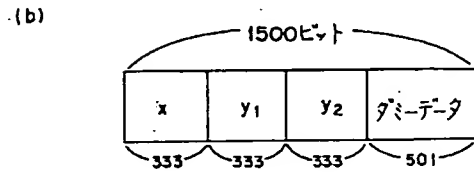
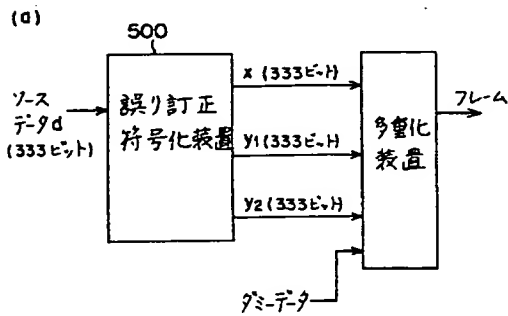
図 20 に示した誤り訂正符号化装置  
の変形例のブロック図





【図22】

(a)は図20に示す誤り訂正符号化装置を使用してソースデータを符号化して固定長のフレームに格納する場合の処理を説明する図。(b)はフレームに格納されるデータを模式的に表した図



【図23】

(a)は図21に示す誤り訂正符号化装置を使用してソースデータを符号化して固定長のフレームに格納する場合の処理を説明する図。(b)はフレームに格納されるデータを模式的に表した図

